

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a computer system that supports a visual user interface designer and a serialization engine capable of saving state for user interface objects created within the visual user interface designer, a method of automatically generating and tracking undo information for changes made to a user interface object within the visual user interface designer so that a developer of the user interface object need not be responsible for generating and tracking the undo information, the method comprising acts of:

receiving one or more change notifications generated in response to one or more changes to the user interface object within the visual user interface designer;

calling the serialization engine to serialize at least a portion of the user interface object into a serialized format that is suitable for representing the one or more changes;

~~creating an~~ using a transaction to fabricate an undo unit from the serialized format of the one or more changes to the user interface object, the transaction spanning the one or more changes and the one or more change notifications; and

adding the undo unit to an undo stack such that the undo unit can subsequently be accessed to perform at least one of undoing and redoing any of the one or more changes.

2. (Original) A method as recited in claim 1, wherein the user interface object comprises a third-party user interface object for use within the visual user interface designer.

3. (Original) A method as recited in claim 1, wherein the one or more change notifications comprise at least one of an adding notification and an added notification.

4. (Original) A method as recited in claim 1, wherein the undo unit identifies one or more routines for undoing the one or more changes made to the user interface object.

5. (Original) A method as recited in claim 4, further comprising acts of:
 - receiving a notification to undo the one or more changes in the undo unit;
 - removing the undo unit from the undo stack; and
 - calling the one or more routines for undoing the one or more changes made to the user interface object.
6. (Original) A method as recited in claim 5, further comprising an act of adding the undo unit to a redo stack as a redo unit.
7. (Original) A method as recited in claim 6, wherein the undo unit identifies one or more routines for redoing the one or more changes made to the user interface object, the method further comprising acts of:
 - receiving a notification to redo the one or more changes in the redo unit;
 - removing the redo unit from the redo stack; and
 - calling the one or more routines for redoing the one or more changes made to the user interface object.
8. (Original) A method as recited in claim 1, wherein the undo unit comprises a name of the user interface object, a type of the user interface object, and one or more previous states of the user interface object prior to the one or more changes.

9. (Currently Amended) ~~For a~~ A computer program product for use at computer system that supports a visual user interface designer and a serialization engine capable of saving state for user interface objects created within the visual user interface designer, ~~a~~ the computer program product comprising one or more computer readable media ~~carrying~~ having stored thereon computer-executable instructions that, when executed at the computer system, cause the computer system to implement a method of automatically generating and tracking undo information for changes made to a user interface object within the visual user interface designer so that a developer of the user interface object need not be responsible for generating and tracking the undo information, including performing the following~~the method comprising acts of:~~

~~an undo engine~~ receiving one or more change notifications generated in response to one or more changes to the user interface object within the visual user interface designer;

~~the undo engine~~ calling the serialization engine to serialize at least a portion of the user interface object into a serialized format that is suitable for representing the one or more changes; and

~~the undo engine creating using a transaction to fabricate an undo unit from the serialized format of the one or more changes to the user interface object, the transaction spanning the one or more changes and the one or more change notifications;~~ and

adding the undo unit to an undo stack such that the undo unit can subsequently be accessed to perform at least one of undoing and redoing any of the one or more changes.

10. (Original) A computer program product as recited in claim 9, wherein at least one of the one or more change notifications initiates a transaction for multiple changes to be made to the user interface object.

11. (Original) A computer program product as recited in claim 9, wherein the one or more change notifications comprise at least one of a changing notification and a changed notification.

12. (Original) A computer program product as recited in claim 9, wherein the user interface object comprises a dialog with an OK button to accept any changes made within the dialog and a CANCEL button to reject any changes made within the dialog.

13. (Original) A computer program product as recited in 12, wherein at least one of the one or more change notifications initiates a transaction for changes to be made to the dialog.

14. (Original) A computer program product as recited in claim 13, the method further comprising acts of:

receiving a notification that the CANCEL button was selected; and
canceling the transaction.

15. (Original) A computer program product as recited in claim 13, the method further comprising acts of:

receiving a notification that the OK button was selected; and
adding the undo unit to an undue stack, wherein the undo unit comprises the transaction.

16. (Original) A computer program product as recited in claim 9, wherein the undo unit comprises a name of the user interface object, a type of the user interface object, and one or more subsequent states of the user interface object after the one or more changes.

17. (Original) A computer program product as recited in claim 9, wherein the user interface object does not implement undo for one or more properties of the user interface object.

18. (Currently Amended) In a computer system that supports a visual user interface designer and a serialization engine capable of saving state for user interface objects created within the visual user interface designer, a method of automatically generating and tracking undo information for changes made to a user interface object within the visual user interface designer so that a developer of the user interface object need not be responsible for generating and tracking the undo information, the method comprising steps for:

an undo engine processing one or more change notifications for one or more changes to the user interface object within the visual user interface designer;

the undo engine persisting initial data that represents an initial state of the user interface object prior to the one or more changes;

the undo engine persisting subsequent data that represents a subsequent state of the user interface object after the one or more changes; and

the undo engine ~~preparing~~ using a transaction to fabricate an undo unit from the initial data and subsequent data for undoing the one or more changes to the user interface object, the transaction spanning the initial data and subsequent data.

19. (Original) A method as recited in claim 18, wherein the one or more change notifications comprise at least one of a removing notification and a removed notification.

20. (Original) A method as recited in claim 18, wherein the undo unit identifies one or more routines for undoing the one or more changes made to the user interface object.

21. (Original) A method as recited in claim 20, further comprising steps for:

storing the undo unit;

listening for a notification to undo the one or more changes in the undo unit;

using the undo unit and the one or more undo routines to undo the one or more changes made to the user interface object; and

deleting the undo unit.

22. (Original) A method as recited in claim 21, further comprising a step for storing the undo unit as a redo unit.

23. (Original) A method as recited in claim 22, wherein the redo unit comprises one or more routines for redoing the one or more changes made to the user interface object, the method further comprising steps for:

- listening for a notification to redo the one or more changes in the redo unit;
- using the redo unit and the one or more redo routines to redo the one or more changes made to the user interface object; and
- deleting the redo unit.

24. (Original) A method as recited in claim 23, wherein the user interface object does not implement redo.

25. (Original) A method as recited in claim 23, wherein the step for the undo engine persisting initial data and persisting subsequent data comprises an act of calling the serialization engine to serialize at least a portion of the user interface object into a serialized format that is suitable for representing the one or more changes.

26. (Currently Amended) ~~For a~~ A computer program product for use at computer system that supports a visual user interface designer and a serialization engine capable of saving state for user interface objects created within the visual user interface designer, ~~a the~~ computer program product comprising one or more computer readable media ~~carrying~~ having stored thereon computer executable instructions that, when executed by a processor, cause an undo engine at the computer system implement a method of automatically generating and tracking undo information for changes made to a user interface object within the visual user interface designer so that a developer of the user interface object need not be responsible for generating and tracking the undo information, including performing the following~~the method comprising steps for:~~

processing one or more change notifications for one or more changes to the user interface object within the visual user interface designer;

persisting an initial data that represents an initial state of the user interface object prior to the one or more changes;

persisting a subsequent data that represents a subsequent state of the user interface object after the one or more changes; and

~~preparing using a transaction to fabricate an~~ undo unit from the initial data and subsequent data for undoing the one or more changes to the user interface object, the transaction spanning the initial data and the subsequent data.

27. (Original) A computer program product as recited in claim 26, wherein the undo unit comprises a transaction for a plurality of changes made to the user interface object.

28. (Original) A computer program product as recited in claim 26, wherein the one or more change notifications comprise a rename notification.

29. (Original) A computer program product as recited in claim 26, wherein the user interface object comprises a dialog with an OK button to accept any changes made within the dialog and a CANCEL button to reject any changes made within the dialog.

30. (Original) A computer program product as recited in claim 29, wherein at least one of the one or more change notifications initiates a transaction for changes to be made to the dialog.

31. (Original) A computer program product as recited in claim 30, the method further comprising acts of:

receiving a notification that the CANCEL button was selected; and
canceling the transaction.

32. (Original) A computer program product as recited in claim 30, the method further comprising acts of:

receiving a notification that the OK button was selected; and
adding the undo unit to an undue stack, wherein the undo unit comprises the transaction.

33. (Original) A computer program product as recited in claim 26, wherein the user interface object is a third-party user interface object for use within the visual user interface designer.

34. (Original) A computer program product as recited in claim 26, wherein the undo unit comprises a name of the user interface object, a type of the user interface object, one or more previous states of the user interface object prior to the one or more changes, and one or more subsequent states of the user interface object after the one or more changes.

35. (Currently Amended) A computer program product for use at a computer system, the computer program product comprising one or more computer readable media ~~carrying~~ having stored thereon computer executable instructions that implement comprising an undo engine that automatically generates and tracks undo information for changes made to a user interface object within a visual user interface designer so that a developer of the user interface object need not generate and track the undo information, the ~~computer program product~~ undo engine comprising:

one or more routines that capture change notifications that are generated automatically in response to changes made to the user interface object within the visual user interface designer;

one or more routines that request state data for the user interface object to use in representing one or more changes to the user interface object;

one or more routines ~~that prepare~~ that use a transaction to fabricate an undo unit from the state data that corresponds to the user interface object, fabricated transactions spanning the changes made to the user interface object and the captured change notifications; and

one or more routines that request storage of the undo unit such that the undo unit can subsequently be accessed to perform at least one of undoing and redoing any of the changes made to the user interface object.

36. (Original) A computer program product as recited in claim 35, further comprising one or more routines for undoing the one or more changes made to the user interface object.

37. (Original) A computer program product as recited in claim 35, further comprising one or more routines for redoing the one or more changes made to the user interface object.

38. (Original) A computer program product as recited in claim 35, further comprising one or more routines for storing the undo unit.

39. (Original) A computer program product as recited in claim 35, further comprising one or more routines that generate the state data for the user interface object.

40. (Original) A computer program product as recited in claim 35, wherein the one or more routines that request state data for the user interface object are capable of interacting with a plurality of distinct state data implementations.

41. (Original) A computer program product as recited in claim 35, wherein the one or more routines that request storage of the undo unit are capable of interacting with a plurality of distinct storage implementations.